

explain_nanosleep(3)

explain_nanosleep(3)

NAMEexplain_nanosleep – explain *nanosleep*(2) errors**SYNOPSIS**

```
#include <libexplain/nanosleep.h>

const char *explain_nanosleep(const struct timespec *req, struct timespec *rem);
const char *explain_errno_nanosleep(int errnum, const struct timespec *req, struct timespec *rem);
void explain_message_nanosleep(char *message, int message_size, const struct timespec *req, struct
timespec *rem);
void explain_message_errno_nanosleep(char *message, int message_size, int errnum, const struct time-
spec *req, struct timespec *rem);
```

DESCRIPTION

These functions may be used to obtain explanations for errors returned by the *nanosleep*(2) system call.

explain_nanosleep

```
const char *explain_nanosleep(const struct timespec *req, struct timespec *rem);
```

The **explain_nanosleep** function is used to obtain an explanation of an error returned by the *nanosleep*(2) system call. The least the message will contain is the value of `strerror(errno)`, but usually it will do much better, and indicate the underlying cause in more detail.

The *errno* global variable will be used to obtain the error value to be decoded.

req The original req, exactly as passed to the *nanosleep*(2) system call.

rem The original rem, exactly as passed to the *nanosleep*(2) system call.

Returns: The message explaining the error. This message buffer is shared by all libexplain functions which do not supply a buffer in their argument list. This will be overwritten by the next call to any libexplain function which shares this buffer, including other threads.

Note: This function is **not** thread safe, because it shares a return buffer across all threads, and many other functions in this library.

Example: This function is intended to be used in a fashion similar to the following example:

```
if (nanosleep(req, rem) < 0)
{
    fprintf(stderr, "%s\n", explain_nanosleep(req, rem));
    exit(EXIT_FAILURE);
}
```

The above code example is available pre-packaged as the *explain_nanosleep_or_die*(3) function.

explain_errno_nanosleep

```
const char *explain_errno_nanosleep(int errnum, const struct timespec *req, struct timespec *rem);
```

The **explain_errno_nanosleep** function is used to obtain an explanation of an error returned by the *nanosleep*(2) system call. The least the message will contain is the value of `strerror(errno)`, but usually it will do much better, and indicate the underlying cause in more detail.

errnum The error value to be decoded, usually obtained from the *errno* global variable just before this function is called. This is necessary if you need to call **any** code between the system call to be explained and this function, because many libc functions will alter the value of *errno*.

req The original req, exactly as passed to the *nanosleep*(2) system call.

rem The original rem, exactly as passed to the *nanosleep*(2) system call.

Returns: The message explaining the error. This message buffer is shared by all libexplain functions which do not supply a buffer in their argument list. This will be overwritten by the next call to any libexplain function which shares this buffer, including other threads.

Note: This function is **not** thread safe, because it shares a return buffer across all threads, and many other functions in this library.

Example: This function is intended to be used in a fashion similar to the following example:

```
if (nanosleep(req, rem) < 0)
{
    int err = errno;
```



explain_nanosleep(3)

explain_nanosleep(3)

```

        fprintf(stderr, "%s\n", explain_errno_nanosleep(err, req,
rem));
        exit(EXIT_FAILURE);
    }

```

The above code example is available pre-packaged as the *explain_nanosleep_or_die(3)* function.

explain_message_nanosleep

```
void explain_message_nanosleep(char *message, int message_size, const struct timespec *req, struct
timespec *rem);
```

The **explain_message_nanosleep** function is used to obtain an explanation of an error returned by the *nanosleep(2)* system call. The least the message will contain is the value of `strerror(errno)`, but usually it will do much better, and indicate the underlying cause in more detail.

The *errno* global variable will be used to obtain the error value to be decoded.

message The location in which to store the returned message. If a suitable message return buffer is supplied, this function is thread safe.

message_size

The size in bytes of the location in which to store the returned message.

req The original req, exactly as passed to the *nanosleep(2)* system call.

rem The original rem, exactly as passed to the *nanosleep(2)* system call.

Example: This function is intended to be used in a fashion similar to the following example:

```

if (nanosleep(req, rem) < 0)
{
    char message[3000];
    explain_message_nanosleep(message, sizeof(message), req,
rem);
    fprintf(stderr, "%s\n", message);
    exit(EXIT_FAILURE);
}

```

The above code example is available pre-packaged as the *explain_nanosleep_or_die(3)* function.

explain_message_errno_nanosleep

```
void explain_message_errno_nanosleep(char *message, int message_size, int errnum, const struct time-
spec *req, struct timespec *rem);
```

The **explain_message_errno_nanosleep** function is used to obtain an explanation of an error returned by the *nanosleep(2)* system call. The least the message will contain is the value of `strerror(errno)`, but usually it will do much better, and indicate the underlying cause in more detail.

message The location in which to store the returned message. If a suitable message return buffer is supplied, this function is thread safe.

message_size

The size in bytes of the location in which to store the returned message.

errnum The error value to be decoded, usually obtained from the *errno* global variable just before this function is called. This is necessary if you need to call **any** code between the system call to be explained and this function, because many libc functions will alter the value of *errno*.

req The original req, exactly as passed to the *nanosleep(2)* system call.

rem The original rem, exactly as passed to the *nanosleep(2)* system call.

Example: This function is intended to be used in a fashion similar to the following example:

```

if (nanosleep(req, rem) < 0)
{
    int err = errno;
    char message[3000];
    explain_message_errno_nanosleep(message, sizeof(message),
err, req, rem);
    fprintf(stderr, "%s\n", message);
}

```



`explain_nanosleep(3)``explain_nanosleep(3)`

```
        exit(EXIT_FAILURE);  
    }
```

The above code example is available pre-packaged as the *explain_nanosleep_or_die(3)* function.

SEE ALSO*nanosleep(2)*

high-resolution sleep

explain_nanosleep_or_die(3)

high-resolution sleep and report errors

COPYRIGHT

libexplain version 1.4

Copyright © 2013 Peter Miller

